Title: Large-Scale Simulations of Sky Surveys

Author(s): Heitmann, Katrin; Habib, Salman; Finkel, Hal; Frontiere, Nicholas; Pope, Adrian; Morozov, Vitali; Rangel, Steve; Kovacs, Eve; Kwan, Julianna; Li, Nan; Rizzi, Silvio; Insley, Joe; Vishwanath, Venkatram; Daniel, David John; Fasel, Patricia Kroll; Zagaris, George

Intended for: Computing in Science and Engineering

Issued: 2017-08-16

Large-Scale Simulations of Sky Surveys

Katrin Heitmann, Salman Habib, Hal Finkel, Nicholas Frontiere, Adrian Pope, Vitali Morozov, Steve Rangel, Eve Kovacs, Juliana Kwan, Nan Li, Silvio Rizzi, Joe Insley, Venkatram Vishwanath, and Tom Peterka | Argonne National Laboratory
David Daniel and Patricia Fasel | Los Alamos National Laboratory
George Zagaris | Kitware

Large-volume sky surveys have accessed the Universe's vast temporal and spatial expanse via a remarkable set of measurements. Interpretation of these cosmological observations requires large-scale numerical simulation and modeling. Addressing analysis workflow complexity is as important as running the underlying extreme-scale simulations. Here, the authors discuss how the Hardware/Hybrid Accelerated Cosmology Code framework addresses these challenges.

Cosmologists have looked deeply at the Universe and found it to be "dark." Over the past three decades, detailed observations carried out from both the ground and from space have spanned the full range of the electromagnetic spectrum—from gamma rays to the radio sky—and have persuasively suggested an astonishing picture:

- approximately 70 percent of the Universe's matter–energy content is made up of a mysterious "dark energy" component;
- this energy is potentially responsible for the Universe's accelerated expansion; and
- 25 percent of the Universe's matter exists in the form of an as-yet unidentified "dark matter" component, while only 0.4 percent of the remaining ordinary matter happens to be visible.

Understanding the mysterious dark sector's physics is the foremost challenge in cosmology today. Major cosmological missions are both ongoing and planned with the goal of creating evermore detailed maps of how mass is distributed in the Universe. These maps hold the key to advancing our knowledge of the Universe's make-up and evolution, enabling us to unlock its "dark" secrets.

Unlike a science based on the experimental method, cosmology lacks investigations under the researcher's control, performed under strict isolation, and allowing step-by-step progress towards solving a physical problem, however complex it may be. The task instead is to make a number of robust observations in which statistical and systematic errors can be bounded, and then to arrive at scientifically defensible inferences about the Universe. To do this, researchers create model universes allowing for different cosmological models and astrophysical effects, mimicking possible observational systematics and even implementing the "clean up" of observational data from unwanted foregrounds that might obscure the signals being searched for. The task's complexity leads inexorably to the use of the world's largest supercomputers.

This requires an end-to-end computational approach, starting from the fundamental theory (Einstein's general relativity and modifications thereof, quantum mechanics), and then to simulating the formation of large-scale structures in the Universe and creating synthetic maps for the target observation, and finally to modeling the instrument and effects that can bias observations, such as atmospheric turbulence. (A detailed discussion of an end-to-end pipeline for the Large Synoptic Survey Telescope is available elsewhere.[1]) We must understand the observed system as a whole, and determine which physics will have important effects on what scales and where simple modeling will suffice, as compared to fully self-consistent simulations, and how uncertainties in modeling and simulations can bias the conclusions. This task is complicated further by the fact that we often

1521-9615/14/$31.00 © 2014 IEEE Copublished by the IEEE CS and the AIP

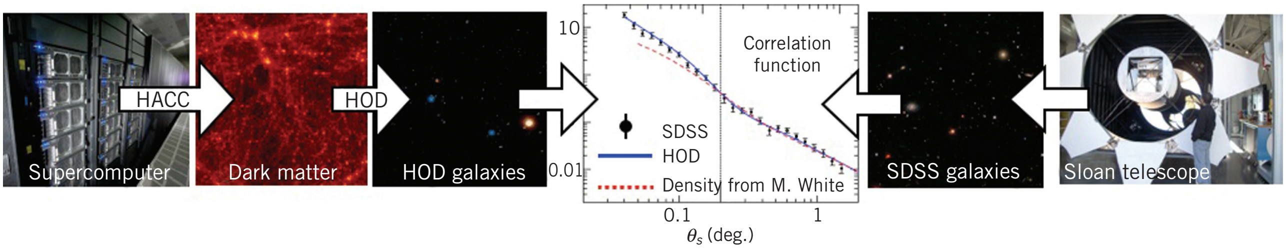

Figure 1. Pipeline to extract cosmological information from galaxy surveys. The halo occupancy distribution (HOD) is a statistical method used to "paint" galaxies onto the dark matter distribution. The mass distribution from large simulations is populated with galaxies that live in dark matter clumps, called "halos." The galaxy's count and brightness is correlated with the halo mass. The results are compared to the galaxy distribution as measured by cosmological surveys.

cannot directly observe what we want to study, but rather must draw conclusions from an indirect analysis, such as by using baryonic tracers (galaxies) of the large-scale structure of the Universe.

Here, we describe our current efforts to create synthetic galaxy maps from large-scale simulations for optical surveys, setting aside telescope modeling as a separate problem. To build these synthetic maps, we must simulate the evolution of the mass distribution in large cosmological volumes with exquisite resolution. To demonstrate the challenge's size, a quick summary of the relevant scales is as follows. Modern survey depths require covering simulation volumes of the order of tens of cubic gigaparsecs (Gpc; 1 pc = 3.26 light years); to follow bright galaxies, structures with a minimum mass of 10^{11} $M_\odot$ ($M_\odot$ = 1 solar mass) must be tracked and resolved by at least a 100 simulation particles. The force resolution (approximately a kiloparsec) must be small compared to the target object's size. This immediately implies a dynamic range (a ratio of smallest resolved scale to box size) of a part in approximately 10^6 (that is, Gpc/kpc) everywhere in the *entire* simulation volume. In terms of the number of simulation particles required, the implied counts range from hundreds of billions to many trillions. This requires access to very large supercomputers and, in today's landscape of diverse supercomputing architectures, a highly scalable and portable N-body code. To this end, we developed the Hardware/Hybrid Accelerated Cosmology Code (HACC), a high-performance code framework targeted at current and future architectures.

The Universe's mass distribution is probed indirectly, as most of the mass is dark and neither emits nor absorbs light. Because light's presence or absence is what we observe, the connection between mass and light is of fundamental importance. To investigate this connection, we need both a major analysis suite and a seamless workflow to ingest the raw simulation output and produce large-scale maps of galaxies. We have created such an analysis environment, which combines in situ analysis tools with a suite of postprocessing steps (described later). As an example, Figure 1 summarizes the analysis path for the statistics of galaxy surveys. In this article, we focus on describing the different steps on the image's left side—that is, how to use large-scale supercomputers to create detailed maps of our Universe as seen through optical telescopes. We also show some concrete examples of our efforts.

HACC Overview

HACC was initially designed for the Roadrunner supercomputer,[2,3] the first system to break the petaflops barrier. With its novel architecture of acceleration via the Cell processor, Roadrunner was by far the most forward-looking machine of its generation, providing a glimpse of the current frontier and some illumination of the path to the exascale.[4]

A modern high-performance code's design must begin with an awareness that methods and algorithms should not be developed without an understanding of future programming paradigms and computing and storage architectures. The HACC computational strategy is based on a hybrid representation of physical information on computational grids as well as "particles" that, depending on the context, can be viewed as tracers of mass, or microfluid elements. This hybrid representation is flexible and maps well to machine architectures, as well as align with multiple programming paradigms. It also provides a broad choice of methods that can be optimized given architectural, power, and other constraints, letting researchers pick the best combination for their particular platform.

Technically speaking, HACC simulates cosmic structure formation by solving the gravitational

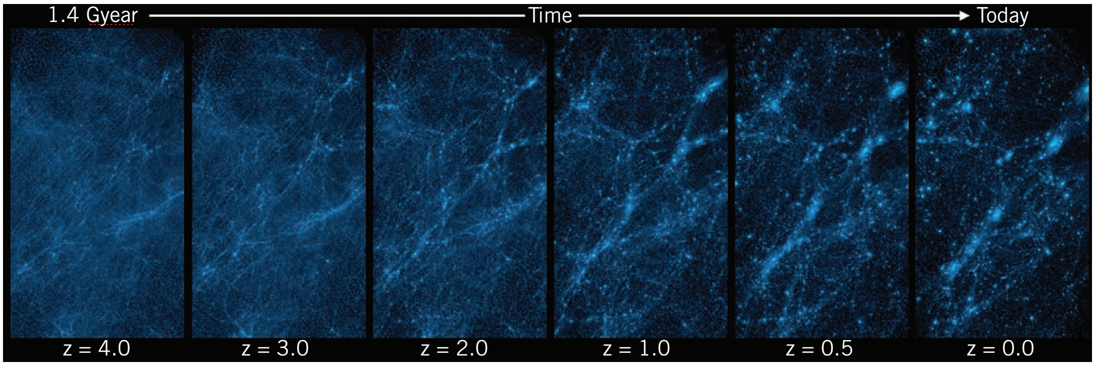

Figure 2. Time evolution of structure formation in a dense region. The zoomed-in frames depict the structure at different redshifts or times, starting 1.4 gigayears (Gyears) after the Big Bang. The images were generated using the vl3 parallel volume rendering system.[6]

Vlasov-Poisson equation in an expanding Universe.[5] The simulation starts from a smooth Gaussian random field that evolves into a "cosmic web" comprised of sheets, filaments, and halos. Figure 2 shows an image of cosmic structure formation over time. The Vlasov-Poisson equation is hopeless to solve as a partial differential equation because of its high dimensionality and the development of nonlinear structure; we thus employ N-body methods.

HACC uses a combination of grid and particle methods: the grid methods resolve the large- to medium- length (smooth) scales and the particle methods resolve the smaller scales. This split between the long- and short-range solver offers a convenient code organization: the long-range force computation—in this case, a *fast Fourier transform-based* (FFT) solver—exists at the code's higher level and is essentially architecture-independent. It is implemented in C/C++/MPI, and its performance and scaling is dominated by the FFT implementation. We have developed a new pencil-decomposed FFT and demonstrated scaling up to 1,572,864 cores on Sequoia, a 96-rack IBM Blue Gene/Q (BG/Q) system.[7] The particle-based short-range solver exists at a lower level of the computational hierarchy and is architecture-tunable. It combines MPI with a variety of local programming models (OpenCL, OpenMP, and CUDA) to readily adapt to different platforms.

To enhance its flexibility, the short-range solver uses a range of algorithms: direct particle–particle interactions (that is, a P^3M algorithm[8]) as on Roadrunner and Titan, or both tree and particle-particle methods as on the IBM BG/Q (a "PPTreePM" algorithm). The grid is responsible for four orders of magnitude of dynamic range, while the particle methods handle the critical two orders of magnitude at the shortest scales, in which particle clustering is maximal and the bulk of the time-stepping computation takes place. An in-depth description of the HACC design and implementation, including the long-range solver, the different short-range solvers, the time stepper, and the code's spatial decomposition and scaling properties, is available elsewhere.[9]

HACC's multi-algorithmic structure attacks several weaknesses of conventional particle codes, including limited vectorization, indirection, complex data structures, lack of threading, and short interaction lists. Currently, HACC is implemented on conventional and Cell/GPU-accelerated clusters[2,3,9] on the IBM BG architecture,[7] and is running on prototype Intel Xeon Phi hardware. HACC is the first—and currently, the only—large-scale cosmology code suite worldwide that can run at scale on *all* available supercomputer architectures. HACC achieved outstanding performance on both Sequoia and Titan, reaching almost 14 pflops (69.2 percent of peak) on Sequoia, a kernel peak of 20.54 pflops on 77 percent of Titan (the full machine was not available for these scaling runs), and 7.9 pflops of sustained performance on 77 percent of Titan for the full code. HACC's outstanding performance and portability has let us carry out some of the challenging simulations needed to advance our understanding of the dark Universe.

Analytics Requirements and Tools

Analyzing large cosmological simulation datasets is as demanding as carrying out the simulations themselves. In fact, some of the computational modeling applied to the outputs in postprocessing is even more complex than the N-body simulation itself with respect to the physical processes involved. To put the analysis task's challenge in context, a single time snapshot from one of the simulations we describe later encompasses 40 Tbytes of raw data, and we must analyze on the order of 100 snapshots. This amount of data clearly demands a carefully considered analysis strategy that combines in situ and postprocessing tools. Another challenge is posed by the fact that the raw data from the simulation is very science-rich. A simulation yields not just optical catalogs (which we describe later), but also field maps, such as the cosmic microwave background temperature, or X-ray flux. It is thus important to store enough of the already-processed data to ensure that new science projects can be carried out later.

To design an efficient workflow to tackle these challenges, and to decide which analysis tools must be run in situ and therefore on the HPC system itself—which means they should scale as well as the main code, a difficult task in and of itself—it is useful to divide the data into three more or less-distinct levels:

- Level I—the raw simulation output, which includes particles, densities, and so on.
- Level II—the "science" level, which contains output rendered as a description useful for further theoretical analysis, including halo and subhalo information, merger trees, and line-of-sight skewers.
- Level III—the "galaxy catalog" level, where the data are further reduced such that they can be interacted with in real time.

Very roughly speaking, at each higher level, the data size reduces from the previous level by two or three orders of magnitude.

The data layer plays a crucial role for science applications. Because of the imbalances in the I/O bandwidth relative to peak performance for the computation and the extreme stressing of file systems, dumping raw data into a storage system for post-analysis is a poor strategy for a problem in which intensive analysis of very large datasets is essential. Therefore, we carry out as much of the raw Level I data analysis as possible on the HPC system itself, and also reduce as much Level I data to Level II as we can. The Level II datasets can then be loaded into an analysis cluster and further analyzed. Figure 3 shows a schematic of the different data levels and analysis hierarchy.

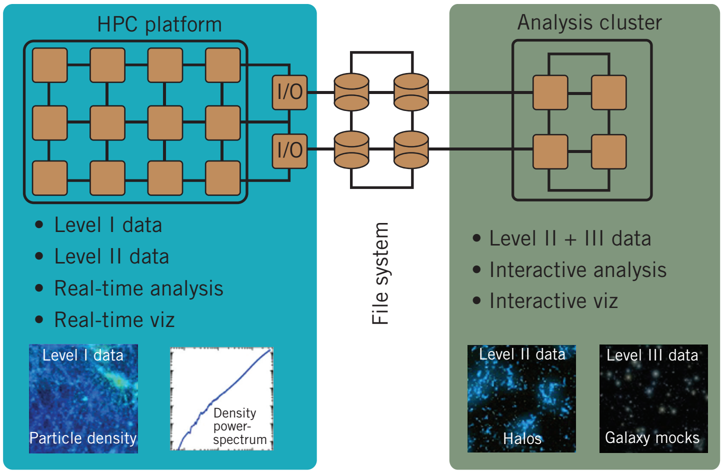

Figure 3. Data levels and analysis hierarchy of a cosmological simulation.

Level I analysis requires algorithms for tasks such as halo-finding, determining correlation functions and a host of other statistical measures, building halo merger trees, and carrying out automated data subsampling. The overall data hierarchy must account for the needs of the analysis routines and the simulation code to maintain locality and avoid data movement. Level II data products can be used for science directly or used to produce Level III data products, such as mock survey catalogs that include galaxies with realistic colors, luminosities, and morphologies. The computational algorithms we apply to address our science goals include density estimation, anomaly detection, tracking, high-dimensional model fitting, and nonparametric inversion. These techniques are computation and memory intensive and have been developed to work within the raw Level I and Level II data products.

We now discuss two concrete examples: the halo finder, which runs in situ with the simulations and reduces data from Level I to Level II; and the halo merger tree code, which acts on Level II data and enables the generation of Level III data.

Halo Finding

The halo concept plays a key role in cosmological simulations. Dark matter halos are the hosts of galaxies, and by mapping out galaxies we can draw conclusions about the dark matter distribution in the Universe. Halos mark over-densities in the dark matter distribution and can be identified through different algorithms. Most commonly, they are found either by locating density peaks directly and growing spheres out to a characteristic over-density or by using neighbor-finding algorithms. Here, we

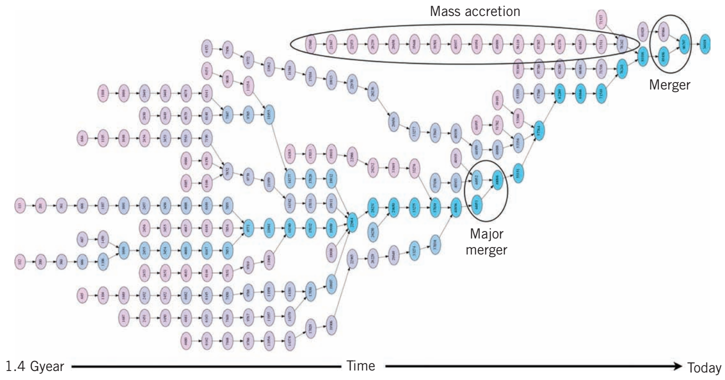

Figure 4. Merger tree for the formation of an individual halo. Each vertex in the tree shows a dark matter halo at a certain time step (time advances from left to right, vertices on each vertical line are halos that exist at the same time). Light colors depict lower mass halos; darker blue colors depict more massive halos. Halos grow over time through two main mechanisms: incremental mass accretion, and the merging of halos; a merger with similar masses is called a "major merger." The merger tree shown here is relatively small. Trees with up to 10,000 nodes can easily exist in the simulations.

will discuss one of the simplest—the *friends-of-friends* (FOF) algorithm, which we used for all of the following results.

In FOF halo finding, for each particle, every particle within a certain distance (the linking length, which is usually between 0.15 to 0.2 of the mean interparticle spacing) is identified as a *friend*. The process is then continued for each friend particle. If the number of particles in such a conglomerate is above a certain threshold (usually approximately 100 particles) the structure is called a *halo*. Its center is found by finding the particles with the most friends (maximum local density); by determining the halo's potential minimum; or by finding the average position from all halo particles (the center of mass). Naively, the FOF algorithm requires N^2 operations, but the algorithm is straightforwardly sped up to NlogN via a tree implementation. Our FOF finder also takes full advantage of the main code's existing overloaded data structure to enable parallel halo finding. Halos can be identified independently on each rank; halos on the edge of a rank are not missed because particle information is available from the neighboring ranks. A final reconciliation step ensures that halos are not counted more than once. Details of the algorithm's implementation and scaling properties are available elsewhere.[10]

As mentioned earlier, the FOF finder reduces the raw Level I simulation data to Level II data. The halo catalog itself, which contains information about halo properties such as position and velocities, is negligible in size compared to the raw data. In addition to the halo catalog, we store the tags of all particles in halos and halo tags, which we need to construct halo merger trees. (Depending on the threshold of what defines a halo, the number of particles in halos is approximately 50 percent of all particles; a halo tag identifies each particle's halo residency). Finally, we store full particle information (positions and velocities) for a subset of particles in halos (usually 1 percent) to enable placements of galaxies at those positions later on, and all particles in halos above a large mass cut-off.

This set of data (halo information and reduced information about particles in halos) defines the set of Level II data connected to the halos and reduces the data volume by a factor of approximately 10. Most of the data is stored in the particle tags of particles that are in halos—once the halo merger trees are built, this information can be discarded and the data reduction then reaches more than a factor of 100, as stated earlier. Halo finding is carried out for roughly 20 percent of all global time steps (no halos exist very early in the simulation). Compared to the time stepper itself, the relative cost of the halo finder decreases over time, but always consumes roughly the same amount of time as a single time step. Because of the data size and computation time consumed, it is infeasible to offload this step to a smaller analysis cluster.

Merger Tree Construction

The FOF algorithm identifies halos from individual snapshots based on the spatial relationship between particles at a fixed point in time. To determine halo temporal evolution, we evaluate the FOF output from the complete sequence of snapshots. Our algorithm compares halos from adjacent snapshots and constructs a graph for representing evolutionary events. The graph, called a *merger tree*, represents each halo by a vertex (see Figure 4) and similar halos in adjacent snapshots with an edge. We define a *similarity measure* as the fraction of shared particles—that is, the particle intersection of two halos—to total particles from the earlier of the two halos.

To construct the merger trees between subsequently taken snapshots, it is sufficient to compare the particle membership functions obtained by the FOF finder. However, computing the pairwise similarity matrix for the halos of all the adjacent snapshots requires some efficiency. To determine multiple-set intersection cardinality, we implement a technique that is linear with respect to the number of particles after an initial particle sort is performed. We utilize a type of sparse matrix representation for the similarity matrix to reduce the otherwise large memory requirement. The memory reduction is significant due to the large amount of sparsity inherent in the problem; in our experience, the computational overhead incurred is marginal.

As mentioned, this approach relies on the result of the FOF algorithm for calculating the halo membership function. Because the clustering algorithm requires halos to have a minimum number of particles, the hard threshold can cause some misidentification of events when halos are near the minimum cutoff value. To reduce these misidentifications, we maintain a windowed history of missing halos. The missing halos' particles are stored for comparison with later snapshots to determine if they re-emerge, and, if so, to be treated as coming from some pre-existing halo.

The Simulated and the Real Universe

We now describe some results from the analysis of recent simulations carried out on Mira at the Argonne Leadership Computing Facility and on Titan at the Oak Ridge Leadership Computing Facility. As mentioned in our introduction, one important task in the analysis is to transform the mass distribution we obtain from the N-body simulations into actual galaxy catalogs. Simulating galaxies from first principles in a cosmological volume is still far from possible—the dynamical range is vast and the physics of galaxy formation is inadequately understood. Instead, galaxies can be painted onto the dark matter distributions, using models of different levels of sophistication. The main assumption here is that "light traces mass"—that is, the galaxies trace the density of the mass distribution, which is predominantly dark. This assumption is true only as an approximation; the aim is to develop more complex prescriptions to "light up" the dark matter distribution with galaxies.

A simple and powerful approach to this problem uses the *Halo Occupation Distribution* (HOD) model.[10] In this approach, "central" and "satellite" galaxies of a certain type are assigned to a dark matter halo depending on the halo's mass. The central galaxy lives at the center of the halo and is the brightest galaxy. If the halo is heavy enough to host more galaxies, satellite galaxies are assigned and placed within the halo. The HOD model is described by approximately five parameters that are tuned to match one observable, such as the galaxy power spectrum. Once the model is fixed, other observables can be predicted from the galaxy catalog.

Recently, we built synthetic sky maps based on a large Mira simulation evolving 32 billion particles in a (2.1 Gpc)^3 volume and investigated the galaxy power spectrum's dependence on the five HOD modeling parameters.[11] Figure 5 shows the best-fit HOD model on top of data from the Baryon Oscillation Spectroscopic Survey (BOSS); as the figure shows, the results from simulations match well with the observational data.

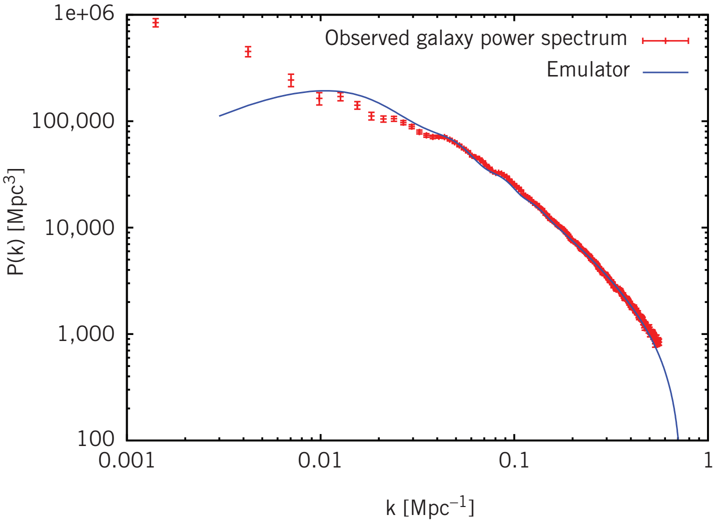

Figure 5. Comparing a simulated galaxy spectrum to observations from the Sloan Digital Sky Survey-Baryon Oscillation Spectroscopic Survey (SDSS-BOSS).[12] The power spectrum is the Fourier analog of the two-point correlation function and characterizes the tendency of galaxies to cluster together.

Although the HOD approach is simple, it has one major shortcoming: it completely neglects a halo's formation history, which will surely carry information about the galaxy population it hosts today. For example, if a halo formed very early and grew mainly through mass accretion, it will not have much star formation today. Or, if the halo underwent a violent merger with another large halo, it will also have a distinct galaxy population. To take these effects into account, researchers have developed *semi-analytic models* (SAMs) that follow each halo's evolution via halo merger trees and, along the way, solve a set of physics equations that approximately describe galaxy formation. SAMs deliver highly detailed descriptions of the galaxies that populate halos, including their colors, positions, and shapes, star formation history, and black hole content. The SAMs' drawback is that they depend on numerous parameters (two to three hundred) that must be tuned to observations.

Figure 6 shows an example of our full simulation and analysis pipeline working to create a

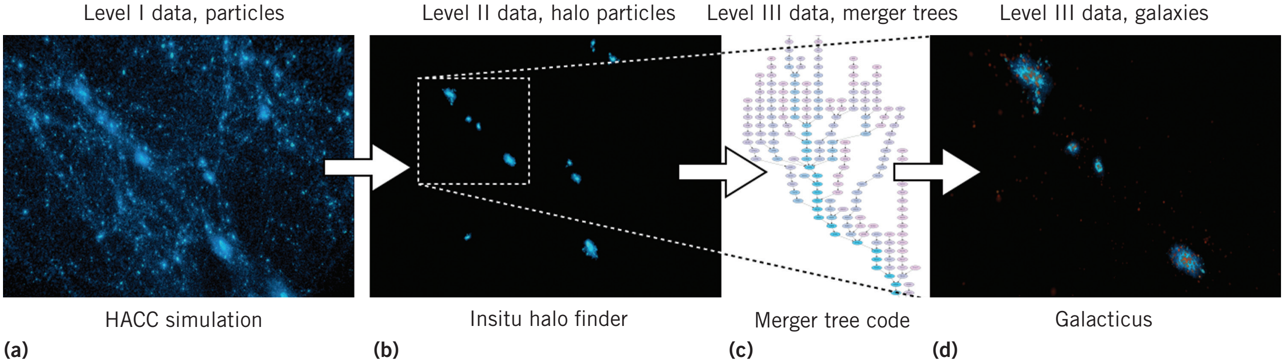

Figure 6. From raw simulation to galaxy catalog. (a) The zoom-in to a full particle distribution from the N-body simulation (Level I data). (b) Dark matter halos identified with the friends-of-friends (FOF) halo finder (Level II data). (c) A merger tree (Level III data). (d) The galaxies embedded in the halos as determined by Galacticus (Level III data). The images were generated using the vl3 parallel volume rendering system.[2]

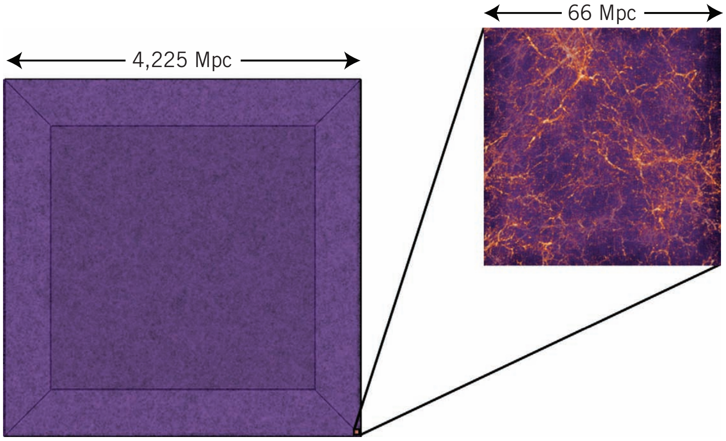

Figure 7. Dynamic range of the Outer Rim simulation. (a) The full simulation volume of (4,225 Mpc)^3. (b) Output from just one of the 524,288 cores.

synthetic galaxy map. The simulation, carried out on Titan, has a mass resolution of approximately 10^9 $M_\odot$ and therefore can capture the smaller halos that host bright galaxies reliably. As the simulation was run, halos were identified on the fly, and the information about particles resident in halos was stored. From this information, merger trees were constructed to track each halo's evolution in detail. Finally, a sophisticated semi-analytic model (Galacticus[13]) was run on the merger trees to generate a full synthetic galaxy sky.

Our last example shows results from the largest high-resolution simulation ever attempted: the Outer Rim simulation. This simulation is currently running on Mira and evolves 1.1 trillion particles in a (4,225 megaparsecs [Mpc])^3 volume. As for the Titan run, each particle has a mass of approximately 10^9 $M_\odot$, but the volume covered is many times larger. The force resolution in the simulation is approximately 4.1 kpc, achieved via a $10{,}240^3$ particle mesh (PM) on the large scales in combination with the tree solver on small scales. To demonstrate the simulation's scale, we show a slice of the full simulation box in Figure 7 as well as the output from one of the 262,144 ranks the simulation is run on.

Already, new science results have been extracted from the simulation (even though it has not quite yet reached the present epoch); Figure 8 shows an example of the exciting science results that are possible. In this case, a halo at a certain time was extracted, a tessellation-based estimator was used to create its 2D projected density, and a ray-tracing code generated a strong gravitational lensing image from a simulated source (Figure 8 shows the workflow). *Strong lensing* refers to the severe distortion of galaxy images and the generation of multiple images due to the presence of a massive intervening object between the source galaxies and the observer. In our case, the halo is the massive object (in the center of Figure 8c), galaxies are placed behind this lens, and the visible arcs are the distorted images as given by a fast ray-tracing algorithm.

The resulting images can be compared to images, such as those taken by the Hubble Space Telescope, yielding new clues about the dark Universe, such as the dark matter properties that make up the lensing halo. ■

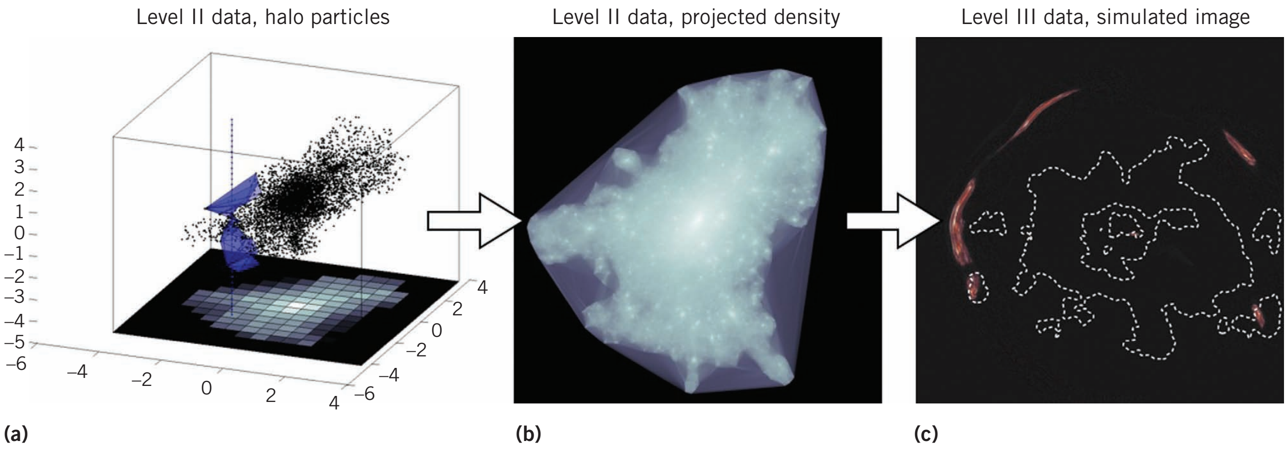
Level II data, halo particles
Level II data, projected density
Level III data, simulated image
4
3
2
1
0
−1
−2
−3
−4
−5
−6
−4
−2
0
2
4
4
2
0
−2
−4
−6
(a)
(b)
(c)

Figure 8. From the raw simulation to a simulated strong lensing image. (a) The tessellation approach for density estimation applied to the particle data extracted with the halo finder. The 2D density field is obtained by a weighted sum of 3D density estimates. The box size is on the scale of an individual halo (in units of Mpc) and the grid resolution is independent of simulation parameters. Points are sampled at discrete intervals on lines normal to the 2D grid cells (shown in blue). Sample points within the tetrahedra intersected by the line are identified and interpolated. (b) A 2D density field is created as seen by an observer—if dark matter were directly visible. (c) Finally, galaxies are placed behind the halo and lensed images of these galaxies are created through a ray-tracing algorithm.

Acknowledgments

The authors were supported by the US Department of Energy, Office of Science, under contract DE-AC02-06CH11357. This research used resources of the Argonne Leadership Computing Facility (ALCF), which is supported by DOE/SC under contract DE-AC02-06CH11357 and resources of the Oak Ridge Leadership Computing Facility (OLCF), which is supported by DOE/SC under contract DE-AC05-00OR22725.

References

1. A. Abate et al., *Large Synoptic Survey Telescope: Dark Energy Science Collaboration*, white paper, 2012.
2. S. Habib et al., “Hybrid Petacomputing Meets Cosmology: The Roadrunner Universe Project,” *J. Physics*: *Conf. Series*, vol. 180, no. 1, 2009, article no. 012019.
3. A. Pope et al., “The Accelerated Universe,” *Computing in Science & Eng.*, vol. 12, no. 4, 2010, pp. 17–25.
4. S. Swaminarayan, “Roadrunner: The Dawn of Accelerated Computing,” *Contemporary High Performance Computing*, CRC Press, 2013, pp. 189–224.
5. K.S. Dolag, A.M. Bykov, and A. Diaferio, “Non-Thermal Processes in Cosmological Simulations,” *Space Science Review*, vol. 134, nos. 1–4, 2008, pp. 311–335.
6. M. Hereld et al., “Exploring Large Data over Wide Area Networks,” *Proc. 2011 IEEE Symp. Large Data Analysis and Visualization*, 2011, pp. 133–134.
7. S. Habib et al., “The Universe at Extreme Scale: Multi-Petaflop Sky Simulation on the BG/Q,” *Proc. Int’l Conf. High Performance Computing, Networking, Storage and Analysis*, 2012; http://arxiv.org/pdf/1211.4864.pdf.
8. R.W. Hockney and J.W. Eastwood, *Computer Simulation Using Particles*, Adam Hilger, 1988.
9. S. Habib et al, “HACC: Extreme Scaling and Performance across Diverse Architectures,” *Proc. Int’l Conf. High Performance Computing, Networking, Storage and Analysis*, 2013, article no. 6.
10. J. Woodring et al., “Analyzing and Visualizing Cosmological Simulations with ParaView,” *The Astrophysical J. Supplement*, vol. 195, no. 1, 2011; doi:10.1088/0067-0049/195/1/11.
11. J. Kwan et al., “Cosmic Emulation: Fast Predictions for the Galaxy Power Spectrum,” preprint, *Astrophysical J.*, 2013.
12. L. Anderson et al., “The Clustering of Galaxies in the SDSS-III Baryon Oscillation Spectroscopic Survey: Baryon Acoustic Oscillations in the Data Release 10 and 11 Galaxy Samples,” *Monthly Notices Royal Astronomical Soc.*, vol. 427, no. 4, 2012, pp. 3435–3467.
13. A. Benson, “Galacticus: A Semi-analytic Model of Galaxy Formation,” *New Astronomy*, vol. 17, no. 2, 2012, pp. 175–197.

Katrin Heitmann is a member of the scientific staff in the High-Energy Physics and Mathematics and Computational Science Divisions at Argonne National Laboratory

and a senior fellow at the Computation Institute and the Kavli Institute for Cosmological Physics at the University of Chicago. Her research interests include physical cosmology, advanced statistical methods, and large-scale computing. Heitmann has a PhD in physics from the University of Dortmund. She is a member of the American Physical Society. Contact her at heitmann@anl.gov.

Salman Habib is a senior scientist in the High-Energy Physics and Mathematics and Computational Science Divisions at Argonne National Laboratory and a senior fellow at the Computation Institute and senior member of the Kavli Institute for Cosmological Physics at the University of Chicago. His research interests include quantum and classical dynamical systems and field theory, including the use of large-scale computing resources to solve problems in these fields, and the application of advanced statistical methods and supercomputing to physical cosmology. Habib has a PhD in physics from the University of Maryland. He is a member of the American Physical Society. Contact him at habib@anl.gov.

Hal Finkel is a computational scientist at Argonne National Laboratory's Leadership Computing Facility. His research interests include theoretical cosmology, numerical algorithms, and compiler technology, focusing on applications requiring large-scale computing. Finkel has a PhD in physics from Yale University. Contact him at hfinkel@anl.gov.

Nicholas Frontiere is a graduate student at the University of Chicago and a researcher at Argonne National Laboratory in the High-Energy Physics Division. His research interests include the application of high-performance computing in large-scale cosmology simulations and the scalability of statistical algorithms in high-performance computing. Frontiere has a BS in physics and mathematics from UCLA. Contact him at nfrontiere@gmail.com.

Adrian Pope is the Arthur Holly Compton postdoctoral fellow in the High-Energy Physics Division at Argonne National Laboratory. His research interests include cosmological measurements from the statistical clustering of galaxies in astronomical sky surveys, as well as astronomical instrumentation, data analysis, parameter estimation, statistical methods, and gravitational N-body simulations on high-performance computing systems. Pope has a PhD in physics and astronomy from Johns Hopkins University. He is a member of the American Astronomical Society. Contact him at apope@anl.gov.

Vitali Morozov is a principal application performance engineer at the Argonne Leadership Computing Facility. His research interests include performance projections and HPC hardware trends, porting and tuning applications (primarily on Blue Gene supercomputers' computer simulations of plasma generation), plasma material interactions, plasma thermal and optical properties, and applications to laser and discharge-produced plasmas. Morozov has a PhD in computer science from Ershov's Institute for Informatics Systems, Novosibirsk, Russia. Contact him at morozov@anl.gov.

Steve Rangel is a doctoral student in the Electrical Engineering and Computer Science Department at Northwestern University. His research interests include large-scale data analysis, high-performance computing, and scalable algorithm design. Contact him at steverangel@gmail.com.

Eve Kovacs is a scientist/programmer in the High-Energy Physics division at Argonne National Laboratory, where she is a member of the Dark Energy Survey (DES) and works with the DES Supernova group on supernova cosmology and systematics analysis. Her research interests include cosmology simulations and semi-analytic galaxy modeling. Kovacs has a PhD in physics from the University of Melbourne. Contact her at kovacs@anl.gov.

Juliana Kwan is a postdoc at Argonne National Laboratory in the High-Energy Physics Division. Her research interests include growth of large-scale structure probes and modelling galaxy distributions using N-body simulations for precision cosmology. Kwan has a PhD in physics from the University of Sydney. Contact her at jkwan@anl.gov.

Nan Li holds a joint postdoctoral position at the University of Chicago and Argonne National Laboratory. His research interests include simulations of gravitational lensing, numerical simulation, and computational cosmology. Li has a PhD in physics from the Chinese Academy of Sciences. Contact him at linan7788626@oddjob.uchicago.edu.

Silvio Rizzi is a postdoctoral appointee in data analysis and visualization at the Argonne National Laboratory's Leadership Computing Facility. His research interests include large-scale data visualization, augmented reality and immersive environments for scientific research, and computer-based medical simulation. Rizzi has a PhD in industrial engineering and operations research from the University of Illinois-Chicago. Contact him at srizzi@anl.gov.

Joe Insley is a principal software development specialist at Argonne National Laboratory and the University of

Chicago. His research interests include the development of parallel and scalable methods for large-scale data analysis and visualization on current and next-generation systems. Insley has an MS in computer science from the University of Illinois at Chicago, and is a senior member of IEEE Computer Society and ACM. Contact him at insley@anl.gov.

Venkatram Vishwanath is an assistant computer scientist in the Mathematics and Computer Science Division at Argonne National Laboratory, where he is a member of the Argonne Leadership Computing Facility; a fellow of the University of Chicago's Computation Institute; and an adjunct professor in the Department of Computer Science at Northern Illinois University. His research interests include runtime and programming models for data-intensive computing, scalable algorithms for data movement, scientific data visualization, and performance analysis for parallel applications. Vishwanath has a PhD in computer science from the University of Illinois at Chicago. Contact him at venkat@anl.gov.

Tom Peterka is an assistant computer scientist at Argonne National Laboratory, a fellow of the University of Chicago's Computation Institute, and an adjunct assistant professor at the University of Illinois. His research interests include large-scale parallelism for in situ analysis of scientific datasets. Peterka has a PhD in computer science from the University of Illinois at Chicago. He is a member of IEEE. Contact him at tpeterka@mcs.anl.gov.

David Daniel is a scientist in the Applied Computer Science group at Los Alamos National Laboratory. His research interests include scientific computing and scalable system software, including cosmology, lattice QCD, and multiphysics simulation codes, and he has made major contributions to MPI libraries including Open MPI and LA-MPI. Daniel has a PhD in theoretical physics from the University of Edinburgh. Contact him at ddd@anl.gov.

Patricia Fasel is a Scientist III in the Information Sciences group at Los Alamos National Laboratory, where she works in algorithm development, software engineering, and data analysis in support of scientific applications. Her research interests include in situ analysis and visualization of large-scale simulations, parallel programming, and large-scale data mining. Fasel has an MS in computer science from Purdue University. Contact her at pkf@anl.gov.

George Zagaris is an R&D engineer in the Scientific Computing Division at Kitware, where he is developing a framework for in situ analysis of large-scale cosmological simulations. His research interests include computational science and computer science in the context of high-performance computing. Zagaris has an MS in computer science from the College of William & Mary. Contact him at george.zagaris@kitware.com.